

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-55135

(43) 公開日 平成10年(1998) 2月24日

(51) Int.Cl. ⁸	識別記号	庁内整理番号	F I	技術表示箇所
G 0 9 C 1/00	6 5 0	7259-5 J	G 0 9 C 1/00	6 5 0 Z
1/10		7259-5 J	1/10	

審査請求 未請求 請求項の数23 O L (全 17 頁)

(21) 出願番号 特願平8-211227

(22) 出願日 平成8年(1996) 8月9日

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番1号

(72) 発明者 北島 弘伸

神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

(72) 発明者 笛木 俊介

神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

(74) 代理人 弁理士 大菅 義之 (外1名)

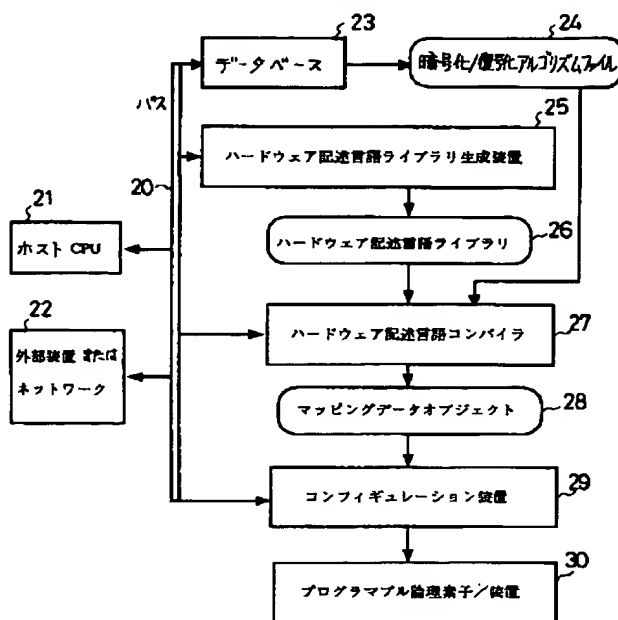
(54) 【発明の名称】 プログラマブルな論理素子／装置を用いた暗号化／復号化装置および方法

(57) 【要約】

【課題】 必要な機密度などの条件に応じて、フレキシブルにアルゴリズムを変更することが可能で、かつ、高速な暗号化／復号化技術を実現することが課題である。

【解決手段】 変更データを受け取ると、ハードウェア記述言語コンパイラ27は、対応する暗号化／復号化アルゴリズムファイル24をデータベース23から取り出し、ハードウェア記述言語ライブラリ生成装置25が生成したハードウェア記述言語ライブラリ26を用いて、それをコンパイルする。コンフィギュレーション装置29は、こうして生成されたマッピングデータオブジェクト28をプログラマブル論理素子／装置30に書き込んで、暗号化／復号化回路を変更する。変更データをもとにして、暗号化／復号化回路の構成が自動的に変更されるので、暗号化／復号化アルゴリズムの変更が容易になる。

暗号化／復号化装置の構成図



1

【特許請求の範囲】

【請求項 1】 少なくとも 1 つ以上のプログラマブル論理素子を含み、該プログラマブル論理素子を用いて、与えられた暗号化の仕様に対応する暗号化回路を生成する回路手段と、

前記暗号化の仕様を変更するための変更データを読み込み、該変更データに基づいて、前記暗号化回路を自動的に変更する変更手段とを備えることを特徴とする暗号化装置。

【請求項 2】 前記変更手段は、前記暗号化回路の構成を表すマッピングデータオブジェクトを前記プログラマブル論理素子に書き込むコンフィギュレーション手段を含み、既存のマッピングデータオブジェクトを前記変更データとして、前記暗号化回路を変更することを特徴とする請求項 1 記載の暗号化装置。

【請求項 3】 前記変更手段は、ハードウェア記述言語により記述されたライブラリをコンパイルして、前記暗号化回路の構成を表すマッピングデータオブジェクトを生成するコンパイラ手段と、該マッピングデータオブジェクトを前記プログラマブル論理素子に書き込むコンフィギュレーション手段とを含み、既存のライブラリを前記変更データとして読み込んでコンパイルし、前記暗号化回路を変更することを特徴とする請求項 1 記載の暗号化装置。

【請求項 4】 前記変更手段は、あらかじめ決められた暗号化のアルゴリズムを記述した暗号化アルゴリズムファイルを記憶するデータベース手段と、ハードウェア記述言語により記述されたライブラリをコンパイルして、前記暗号化回路の構成を表すマッピングデータオブジェクトを生成するコンパイラ手段と、該マッピングデータオブジェクトを前記プログラマブル論理素子に書き込むコンフィギュレーション手段とを含み、外部から前記変更データとして与えられた設定データに基づいて、対応する暗号化アルゴリズムファイルを検索し、該対応する暗号化アルゴリズムファイルに記述されたライブラリを用いて、前記暗号化回路を変更することを特徴とする請求項 1 記載の暗号化装置。

【請求項 5】 通信ネットワークに接続するネットワーク接続手段をさらに備え、前記変更手段は、前記変更データを該ネットワークから読み込むことを特徴とする請求項 1 記載の暗号化装置。

【請求項 6】 前記ネットワーク接続手段は、暗号化された前記変更データを前記ネットワークから受け取り、前記変更手段は、前記暗号化された変更データに基づいて前記暗号化回路を変更することを特徴とする請求項 5 記載の暗号化装置。

【請求項 7】 前記変更手段は、前記暗号化の仕様を定期的に更新することを特徴とする請求項 1 記載の暗号化装置。

【請求項 8】 前記変更手段は、外部からの要請に基づ

2

いて、前記暗号化の仕様を更新することを特徴とする請求項 1 記載の暗号化装置。

【請求項 9】 前記変更手段は、被暗号化データの通信経路、該被暗号化データの機密度、および該被暗号化データに対して要求される処理速度のうち、少なくとも 1 つに応じて、前記暗号化の仕様を変更することを特徴とする請求項 1 記載の暗号化装置。

【請求項 10】 少なくとも 1 つ以上のプログラマブル論理素子を含み、該プログラマブル論理素子を用いて、与えられた復号化の仕様に対応する復号化回路を生成する回路手段と、

前記復号化の仕様を変更するための変更データを読み込み、該変更データに基づいて、前記復号化回路を自動的に変更する変更手段とを備えることを特徴とする復号化装置。

【請求項 11】 前記変更手段は、前記復号化回路の構成を表すマッピングデータオブジェクトを前記プログラマブル論理素子に書き込むコンフィギュレーション手段を含み、既存のマッピングデータオブジェクトを前記変更データとして、前記復号化回路を変更することを特徴とする請求項 10 記載の復号化装置。

【請求項 12】 前記変更手段は、ハードウェア記述言語により記述されたライブラリをコンパイルして、前記復号化回路の構成を表すマッピングデータオブジェクトを生成するコンパイラ手段と、該マッピングデータオブジェクトを前記プログラマブル論理素子に書き込むコンフィギュレーション手段とを含み、既存のライブラリを前記変更データとして読み込んでコンパイルし、前記復号化回路を変更することを特徴とする請求項 10 記載の復号化装置。

【請求項 13】 前記変更手段は、あらかじめ決められた復号化のアルゴリズムを記述した復号化アルゴリズムファイルを記憶するデータベース手段と、ハードウェア記述言語により記述されたライブラリをコンパイルして、前記復号化回路の構成を表すマッピングデータオブジェクトを生成するコンパイラ手段と、該マッピングデータオブジェクトを前記プログラマブル論理素子に書き込むコンフィギュレーション手段とを含み、外部から前記変更データとして与えられた設定データに基づいて、対応する復号化アルゴリズムファイルを検索し、該対応する復号化アルゴリズムファイルに記述されたライブラリを用いて、前記復号化回路を変更することを特徴とする請求項 10 記載の復号化装置。

【請求項 14】 通信ネットワークに接続するネットワーク接続手段をさらに備え、前記変更手段は、前記変更データを該ネットワークから読み込むことを特徴とする請求項 10 記載の復号化装置。

【請求項 15】 前記ネットワーク接続手段は、復号化された前記変更データを前記ネットワークから受け取り、前記変更手段は、前記復号化された変更データに基

3

づいて前記復号化回路を変更することを特徴とする請求項 14 記載の復号化装置。

【請求項 16】 前記変更手段は、前記復号化の仕様を定期的に更新することを特徴とする請求項 10 記載の復号化装置。

【請求項 17】 前記変更手段は、外部からの要請に基づいて、前記復号化の仕様を更新することを特徴とする請求項 10 記載の復号化装置。

【請求項 18】 前記変更手段は、被復号化データの通信経路、該被復号化データの機密度、および該被復号化データに対して要求される処理速度のうち、少なくとも 1 つに応じて、前記復号化の仕様を変更することを特徴とする請求項 10 記載の復号化装置。

【請求項 19】 少なくとも 1 つ以上のプログラマブル論理素子を含み、該プログラマブル論理素子を用いて、与えられた仕様に対応する回路を生成する回路手段と、前記回路の仕様を変更するための変更データであって、暗号化と復号化のいずれか一方の仕様を表す該変更データを読み込み、該変更データに基づいて、前記回路を自動的に変更する変更手段とを備えることを特徴とする暗号処理装置。

【請求項 20】 通信ネットワークを介して暗号化されたデータをやり取りする通信システムのための暗号処理システムであって、
少なくとも 1 つ以上のプログラマブル論理素子を含み、与えられた暗号化の仕様に対応する暗号化回路を生成する暗号化回路手段と、
前記暗号化の仕様を変更するための暗号化変更データを読み込み、該暗号化変更データに基づいて、前記暗号化回路を自動的に変更する暗号化変更手段と、
少なくとも 1 つ以上のプログラマブル論理素子を含み、与えられた復号化の仕様に対応する復号化回路を生成する復号化回路手段と、
前記復号化の仕様を変更するための復号化変更データを読み込み、該復号化変更データに基づいて、前記復号化回路を自動的に変更する復号化変更手段とを備えることを特徴とする暗号処理システム。

【請求項 21】 少なくとも 1 つ以上のプログラマブル論理素子を用いて、与えられた暗号化の仕様に対応する暗号化回路を生成し、
前記暗号化の仕様を変更するための変更データを読み込み、該変更データに基づいて、前記暗号化回路を自動的に変更することを特徴とする暗号化方法。

【請求項 22】 少なくとも 1 つ以上のプログラマブル論理素子を用いて、与えられた復号化の仕様に対応する復号化回路を生成し、
前記復号化の仕様を変更するための変更データを読み込み、該変更データに基づいて、前記復号化回路を自動的に変更することを特徴とする復号化方法。

【請求項 23】 少なくとも 1 つ以上のプログラマブル

4

論理素子を用いて、与えられた仕様に対応する回路を生成し、

前記回路の仕様を変更するための変更データであって、暗号化と復号化のいずれか一方の仕様を表す該変更データを読み込み、該変更データに基づいて、前記回路を自動的に変更することを特徴とする暗号処理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ファイルやメールの暗号化、メッセージやユーザの認証などの情報セキュリティ一般に広く用いられる暗号技術に係り、情報を暗号化する暗号化装置、暗号化された情報を復号化する復号化装置、および暗号化／復号化方法に関する。

【0002】

【従来の技術】現在用いられている暗号には大きく分けて、秘密鍵暗号と公開鍵暗号とがある。ここでは、まずこれらの代表として、現在最も広汎に採用されている DES (Data Encryption Standard) 暗号と RSA (Rivest-Shamir-Adleman) 暗号をそれぞれ例にとり、暗号化のアルゴリズムを説明する。

【0003】まず、DES は、米国を中心として採用されている代表的な秘密鍵暗号化アルゴリズムの規格である。DES の暗号化アルゴリズムでは、数値化された平文 (plaintext) データを例えば 64 ビットの固定長ブロックに分割し、そのブロック単位で秘密鍵を用いたさまざまな演算を行うことで、平文データの暗号化を行う。この秘密鍵は、被暗号化データである平文データと同じビット長である。

【0004】図 16 は、ブロック長が 64 ビットの場合の DES の暗号化アルゴリズムの概要を示している。図 16 において、64 ビットの暗号化鍵は縮約転置 1 を施されて、1 段目の処理に入力される。ここで、縮約転置 1 とは、入力データの 1 部を除いて残りの部分を転置する操作を意味し、転置とは、データの部分的な入れ替え操作を意味する。

【0005】転置された暗号化鍵は前半と後半の 2 つの部分に分割され、それぞれの部分に巡回シフト 2 が施される。巡回シフト 2 とは、入力データを左または右にサイクリックにシフトする操作を意味する。巡回シフト 2 の後、さらに縮約転置 3 が施される。

【0006】また、64 ビットの平文は、転置 4 が施された後、前半と後半の 2 つの部分に分割されて 1 段目の処理に入力される。そして、その片方には、縮約転置 3 の後の暗号化鍵を用いた非線形変換 5 が施されて、加算 6 においてもう一方と加算される。このような処理が m 段目まで繰り返され、m 段目の処理の結果に転置 7 が施されて 64 ビットの暗号文 (cryptogram) となる。

【0007】DES の復号化アルゴリズムも、図 16 の暗号化アルゴリズムとほとんど同じであるが、巡回シフト 2 においては、暗号化アルゴリズムと逆向きにデータ

5

をシフトする必要がある。

【0008】次に、RSAの暗号化アルゴリズムは、非常に強力な公開鍵暗号化アルゴリズムであり、データの暗号化だけでなく、メッセージやユーザの認証も行うことができる。このアルゴリズムでは、公開鍵と秘密鍵の二つの暗号化鍵を使用する。公開鍵は、文書やネットワーク上のデータの形で公開され、誰でもアクセス可能な状態に置かれるが、秘密鍵は、使用者が厳密に保管する必要がある。

【0009】RSAの暗号化アルゴリズムは、数論的な

$$n = p \cdot q$$

$$e \cdot d \equiv 1 \pmod{(p-1)(q-1)}$$

ここで、 p と q は素数である。(2)式は、法 $(p-1)(q-1)$ の下での合同式(congruence)であって、 $e \cdot d$ と1は $(p-1)(q-1)$ を法として合同であることを表している。言い換えれば、 $e \cdot d - 1$ は、 $(p-1)(q-1)$ で割り切れるということである。

$$C \equiv M^e \pmod{n}$$

となる。そして、復号化9においては、暗号文 C は法 n

$$M \equiv C^d \pmod{n}$$

となる。

【0012】このようにして暗号化された暗号文 C を解読するには、秘密鍵 d の値を知る必要があるが、そのためには n を素因数分解して、素数 p と q を求めなければならない。しかし、 n が非常に大きな数の場合、現在の計算機パワーでは、現実的な処理時間内で素因数分解を行うことができない。

【0013】

【発明が解決しようとする課題】しかしながら、従来の暗号化／復号化技術には次のような問題がある。上述のような堅牢性の高い暗号化アルゴリズムは、ビット長の比較的長い暗号化鍵を用いて複雑な演算を行うため、一般に処理速度が遅く、ソフトウェアでの実装は小規模データの処理などに用途が限定されてしまう。特に、ネットワークで結ばれた情報処理装置間で、通信しながら暗号化／復号化を行うようなリアルタイム処理には、ほとんど実用にならない。

【0014】そこで、暗号化アルゴリズムをハードウェア的に実現したチップがすでに販売されているが、使用可能なアルゴリズムや暗号化鍵のビット長などが固定されており、実用上のフレキシビリティに欠けている。

【0015】特に、DESのブロック長や、RSAの暗号化鍵のビット長などは、暗号化アルゴリズムの堅牢性に強く関係している。これらが小さすぎると、巧妙な手法や強力な計算機の援用により、暗号が破られる可能性が高くなる。したがって、セキュリティを確保するためには、機密の程度やその時代の計算機パワーに合わせて、十分な設定値を採用する必要がある。

【0016】本発明の課題は、必要な機密度などの条件に応じて、フレキシブルにアルゴリズムを変更すること

6

演算を暗号化や復号化に用いており、巨大な素数の素因数分解が非常に困難であることを、暗号の堅牢性の基礎に置いている。

【0010】図17は、公開鍵で暗号化したデータを秘密鍵で復号化する場合のRSAの暗号化／復号化アルゴリズムの概要を示している。図17において、暗号化8で用いられる暗号化鍵 (e, n) は、公開された特定の整数 e と n の組であり、復号化9で用いられる復号化鍵 (d, n) は、同じ n と非公開の整数 d の組である。これらの数は、次式に基づいて決められる。

$$(1)$$

$$(2)$$

る。また、 $e < n$ であり、 e と $(p-1)(q-1)$ は互いに素である。

【0011】まず、暗号化8において、平文 M は法 n の下で e 乗され、暗号文 C に変換される。すなわち、

$$(3)$$

の下で d 乗されて、平文 M に戻される。すなわち、

$$(4)$$

が可能で、かつ、高速な暗号化／復号化装置および暗号化／復号化方法を提供することである。

【0017】

【課題を解決するための手段】図1は、本発明の暗号化／復号化装置の原理図である。図1の暗号化／復号化装置は、回路手段11および変更手段12を備える。

【0018】回路手段11は、少なくとも1つ以上のプログラマブル論理素子13を含み、それらのプログラマブル論理素子13を用いて、与えられた暗号化／復号化の仕様に対応する暗号化／復号化回路を生成する。

【0019】変更手段12は、上記暗号化／復号化の仕様を変更するための変更データを読み込み、その変更データに基づいて、上記暗号化／復号化回路を自動的に変更する。

【0020】プログラマブル論理素子13としては、例えばFPGA(field programmable gate array)が用いられ、回路手段11は、プログラマブル論理素子13の仕様を、その内部構成を表すマッピングデータオブジェクトなどの形で読み込むことで、暗号化／復号化の仕様に対応する暗号化／復号化回路を生成する。

【0021】変更手段12は、マッピングデータオブジェクトなどの形で与えられる上記変更データを読み込み、その変更データをプログラマブル論理素子13にダイナミックに反映させることで、暗号化／復号化回路内のゲート配置や配線などを変更する。

【0022】こうして変更された暗号化／復号化回路は、入力される被暗号化データ(平文)／被復号化データ(暗号文)を、変更後の仕様に従って、暗号化データ(暗号文)／復号化データ(平文)に変換する。

【0023】このような暗号化／復号化装置によれば、

7

暗号化／復号化回路の内部構成が可変であるため、データの機密度や用途などに応じて、暗号化／復号化回路の仕様をダイナミックに変更することが可能になる。また、その変更は、与えられた変更データをもとにして、自動的に行われる。

【0024】また、変更手段12にマッピングデータオブジェクトを自動生成する機能を持たせれば、暗号化／復号化アルゴリズムの種類などを変更データとして指定するだけで、暗号化／復号化回路の仕様を自動的に変更することも可能である。

【0025】したがって、特に回路設計の知識を持たないユーザであっても、簡単に暗号化／復号化回路を変更することができ、フレキシビリティに富んだ装置が実現される。さらに、暗号化／復号化の動作自身はハードウェアにより実行されるので、ソフトウェアによる処理に比べてはるかに高速である。

【0026】例えば、図1の回路手段11は、実施形態の図2におけるプログラマブル論理素子／装置30およびその周辺回路（不図示）に対応し、変更手段12は、ホストCPU（中央処理装置）21、ハードウェア記述言語ライブラリ生成装置25、ハードウェア記述言語コンパイラ27、およびコンフィギュレーション装置29

に対応する。

【0027】**【発明の実施の形態】**以下、図面を参照しながら、本発明の実施の形態を詳細に説明する。本発明の暗号化／復号化装置では、主としてFPGAのようなプログラマブル論理素子／装置を用いて、暗号化／復号化アルゴリズムを実装する。ここで、論理素子とは1つの半導体チップを意味し、論理装置とは2つ以上の半導体チップを含むような基板や装置を意味する。また、プログラマブル論理素子／装置とは、ユーザ自身が書き込み機（アクティベータ）と設計ソフトウェアを用いて、短時間で試作できるような論理素子／装置である。

【0028】本発明では、FPGA以外に、FPGAの1／10程度の回路規模を持つPLD（programmable logic device）、PLA（programmable logic array）、ASIC（application specific integrated circuit）など、任意のプログラマブル論理素子／装置を用いることができる。

【0029】プログラマブル論理素子／装置は、ユーザ自身が内部の論理を作成／変更できるため、実現される暗号化／復号化装置の仕様（スペック）は、既存のマッピングデータ、ネットワーク経由、自動生成などの方法によりダイナミックに変更可能となる。このため、データの機密度や用途に応じて、ユーザが暗号化／復号化装置をカスタマイズすることができる。

【0030】このようなシステムを採用することにより、複数のアルゴリズムや複数のブロック長や複数の鍵のビット長に対して、ダイナミックに対応可能な暗号化

8

／復号化装置が実現される。また、その装置本体はハードウェア的に実現されているので、暗号化／復号化の実行時においては、大規模データの処理やリアルタイム処理にも十分な効率が確保できる。

【0031】図2は、このような暗号化／復号化装置の構成を示している。図2の暗号化／復号化装置は、ホストCPU21、データベース23、ハードウェア記述言語ライブラリ生成装置25、ハードウェア記述言語コンパイラ27、コンフィギュレーション装置29、プログラマブル論理素子／装置30、およびこれらの各装置を結ぶバス20を備え、動作モードとしてコンフィギュレーションフェーズと実行フェーズを持っている。

【0032】コンフィギュレーションフェーズにおいて、ユーザから特定の暗号化／復号化回路の作成を指示されると、ハードウェア記述言語コンパイラ27は、まず対応する暗号化／復号化アルゴリズムをハードウェア記述言語で記述した暗号化／復号化アルゴリズムファイル24を、データベース23から取り出す。暗号化／復号化回路作成の指示は、外部装置またはネットワーク22からも受け付けることができる。

【0033】ここで、ハードウェア記述言語（HDL：hardware description language）とは、プログラマブル論理素子／装置30の内部構成を記述するための言語で、VHDL（VHSIC-HDL：very high-speed integrated circuit hardware description language）や、それを改良したVerilog-HDLなどがある。例えば、プログラマブル論理素子／装置30のピン番号やファンクション（ロジック）などが、ハードウェア記述言語により記述される。

【0034】次に、ハードウェア記述言語コンパイラ27は、ハードウェア記述言語ライブラリ生成装置25が生成したハードウェア記述言語ライブラリ26を用いて、暗号化／復号化アルゴリズムファイル24をコンパイルし、マッピングデータオブジェクト28を生成する。

【0035】マッピングデータオブジェクト28は、バイナリデータのビット列から成り、プログラマブル論理素子／装置30の内部のゲート配置や配線などを表す。FPGAを用いた場合、そのテクノロジーに適合した形式のバイナリデータが用いられ、それがFPGAにダウンロードされると、特定の機能が設定される。

【0036】コンフィギュレーション装置29は、マッピングデータオブジェクト28をプログラマブル論理素子／装置30に書き込んで、配線やロジックを形成し、ユーザの指示に対応する特定の暗号化／復号化回路を作成する。

【0037】本実施形態においては、ハードウェア記述言語ライブラリ生成装置25、ハードウェア記述言語コンパイラ27、およびコンフィギュレーション装置29の各機能は、ホストCPU21が実行するプログラムに

より実現される。

【0038】図3は、図2の暗号化／復号化装置を実現する情報処理装置の構成図である。図3の情報処理装置は、CPU31、メモリ32、入力装置33、出力装置34、外部記憶装置35、媒体駆動装置36、ネットワーク接続装置37を備え、それらの各装置はバス38により互いに結合されている。

【0039】CPU31はホストCPU21に対応し、メモリ32に格納されたプログラムを実行して、ハードウェア記述言語ライブラリ生成装置25、ハードウェア記述言語コンパイラ27、およびコンフィギュレーション装置29の各機能を実現する。メモリ32としては、例えばROM (read only memory)、RAM (random access memory) などが用いられる。

【0040】入力装置33は、例えばキーボード、ポインティングデバイスなどに相当し、ユーザからの指示の入力に用いられる。また、出力装置34は、表示装置やプリンタなどに相当し、メッセージや処理結果などの出力に用いられる。

【0041】外部記憶装置35は、例えば、磁気ディスク装置、光ディスク装置、光磁気ディスク装置などであり、プログラムやデータを保存することができる。また、暗号化／復号化アルゴリズムファイル24、ハードウェア記述言語ライブラリ26、マッピングデータオブジェクト28などを保存するデータベース23としても使用される。

【0042】媒体駆動装置36は、可搬記憶媒体39を駆動し、その記憶内容にアクセスすることができる。可搬記憶媒体39としては、メモ리카ード、フロッピーディスク、CD-ROM (compact disk read only memory)、光ディスク、光磁気ディスクなど、任意の計算機読み出し可能記憶媒体を使用することができる。この可搬記憶媒体39には、データのほかに、図2の暗号化／復号化装置の処理を行うプログラムが格納される。

【0043】ネットワーク接続装置37は、LAN (local area network) などの任意の通信ネットワークに接続され、通信に伴うデータ変換等を行う。暗号化／復号化装置は、ネットワーク接続装置37を介して、外部の情報処理装置などからデータやプログラムを受け取ることができる。

【0044】例えば、DESの暗号化／復号化アルゴリズムを実装する場合、基本ロジックのハードウェア記述言語ライブラリ26として、16／32／64ビット加算器、16／32／64ビット減算器、8／16／32／64／128ビットレジスタ、8／16／32／64／128ビット左右シフトレジスタ、16／32／64／128ビットインクリメントカウンタ、16／32／64／128ビットデクリメントカウンタ、16／32／64ビットDES関数発生器、クロック回路、論理和回路、および論理積回路があらかじめ生成され、データ

ベース23に保存される。

【0045】一例として、16ビットインクリメントカウンタをVeli-log-HDLで記述すると、図4のようになる。図4のハードウェア記述言語ライブラリ26では、クロックの立ち上がりでカウント値qがインクリメントされることが記されている。

【0046】また、RSAの暗号化／復号化アルゴリズムを実装する場合、基本ロジックのハードウェア記述言語ライブラリ26として、16／32／64／128ビット乗算器、16／32／64ビット加算器、16／32／64ビット減算器、8／16／32／64／128ビットレジスタ、16／32／64／128ビットインクリメントカウンタ、16／32／64／128ビットデクリメントカウンタ、クロック回路、論理和回路、および論理積回路があらかじめ生成され、データベース23に保存される。

【0047】図2の暗号化／復号化装置にコンフィギュレーションの指示を与える際、暗号化／復号化アルゴリズムの種類、暗号化／復号化鍵のビット長などの設定データを、コマンドの形式で指定する必要がある。

【0048】例えば、RSAの暗号化の場合、アルゴリズムの種類としてRSAが指定され、暗号化鍵(e, n)のビット長および暗号化鍵の値が設定される。Veli-log-HDLによる記述では、配線のビット幅wireの指定が数値として必要になるので、ホストCPU21は、この数値を設定データから生成して、暗号化アルゴリズムファイル24中のコードに埋め込む。

【0049】図5は、ビット幅の数値が埋め込まれた暗号化アルゴリズムファイル24の例を示している。図5のファイルにおいて、行L1、L2、L3の位置に、それぞれ、平文Mと暗号文Cのビット幅b1=15、暗号化鍵eのビット幅b2=7、暗号化鍵nのビット幅b3=63が記されている。

【0050】ハードウェア記述言語コンパイラ27は、このような配線情報に基づいて選択したハードウェア記述言語ライブラリ26を、暗号化アルゴリズムファイル24の記述に従って合成して、マッピングデータオブジェクト28を生成する。

【0051】次に、図6を参照しながら、コンフィギュレーションフェーズにおける暗号化／復号化回路の作成処理のフローを説明する。図6は、外部から与えられた設定データに基づいて、主としてRSAの暗号化回路を作成する場合のフローチャートであるが、他の暗号化／復号化回路作成の場合も基本的に同様である。

【0052】図6において処理が開始されると、ハードウェア記述言語コンパイラ27は、まず指定された暗号化アルゴリズムの種類と、暗号化鍵のビット長と、暗号化鍵の数値とを設定データとして設定し(ステップS1、S2、S3)、ハードウェア記述言語で記述された対応する暗号化アルゴリズムファイル24を、データベ

ース23から自動的に検索する(ステップS4)。そして、検索した暗号化アルゴリズムファイル24の変数コードに、設定データ的具体値を入力する(ステップS5)。

【0053】次に、ハードウェア記述言語ライブラリ26を利用して、暗号化アルゴリズムファイル24をコンパイルする(ステップS6)、これにより、プログラマブル論理素子/装置30の内部の配置や配線が最適化され、設定データにより指定された特定の暗号化回路のマッピングデータオブジェクト28が生成される(ステップS7)。

【0054】次に、コンフィギュレーション装置29は、プログラマブル論理素子/装置30の周辺回路(不図示)のタイミング信号を発生させ(ステップS8)、プログラマブル論理素子/装置30にマッピングデータオブジェクト28をダウンロードする。こうして、プログラマブル論理素子/装置30の配置や配線が作成される(ステップS9)、処理が終了する。

【0055】このように、コンフィギュレーションフェーズにおいては、簡単な設定データを指定するだけで、プログラマブル論理素子/装置30のプログラミングが自動的に行われるため、設計能力のないユーザでも暗号化/復号化回路を作成することができる。また、設計者にとっても、設計ソフトウェアを用いて回路全体を設計する必要がなく、より短時間で暗号化/復号化回路を作成することができる。

【0056】図6の処理では、外部から与えられた設定データに基づいて暗号化/復号化回路の仕様を自動生成しているが、他の方法で仕様を変更してもよい。例えば、暗号化鍵のビット長などの具体値が既に埋め込まれた暗号化/復号化アルゴリズムファイル24を保存しておき、それに基づいてコンパイルを行ってもよい。

【0057】また、暗号化/復号化アルゴリズムファイル24を利用せずに、データベース23に保存されている既存のハードウェア記述言語ライブラリ26をコンパイルするだけで仕様を変更したり、既存のマッピングデータオブジェクト28を直接ダウンロードしてそれを変更したりすることもできる。さらに、ネットワーク経由で読み込んだハードウェア記述言語ライブラリ26やマッピングデータオブジェクト28を用いて、暗号化/復号化回路の仕様を変更することも可能である。

【0058】設定データ、暗号化/復号化アルゴリズムファイル24、ハードウェア記述言語ライブラリ26、マッピングデータオブジェクト28のように、暗号化/復号化回路の仕様を変更するために使用され得る情報が、上述の変更データに相当する。

【0059】一方、実行フェーズにおいては、図7に示すように、コンフィギュレーション済みのプログラマブル論理素子/装置30は、ホストCPU21またはネットワーク40から平文/暗号文を受け取り、その暗号化

／復号化を行う。そして、得られた暗号文/平文は、ホストCPU21またはネットワーク40に出力される。

【0060】プログラマブル論理素子/装置30は、ネットワーク40との間のデータの入出力を、TCP/IP (transmission control protocol/internet protocol) などのプロトコルにより、直接またはホストCPU21経由で行うことができる。直接データの入出力を行う場合は、プログラマブル論理素子/装置30は、TCP/IP制御用のハードウェア(不図示)を介して、ネットワーク40と接続される。

【0061】図7のプログラマブル論理素子/装置30は、設定データの仕様に適合する平文/暗号文を暗号化/復号化するハードウェア回路であり、ソフトウェアを利用した暗号化/復号化処理に比べて、はるかに高速に暗号化/復号化を実行する。このため、ネットワーク40との間でやり取りされるデータのリアルタイム処理に適している。

【0062】次に、図8から図11までを参照しながら、プログラマブル論理素子/装置30を用いて作成される暗号化/復号化回路の例を説明する。図8および図9は、DESの暗号化回路の例を示している。図8は、制御用のタイミング発生と平文の変換を行う回路部分を示しており、図9は、暗号化鍵の変換を行う回路部分を示している。

【0063】コンフィギュレーションフェーズでは、まず入力/出力のエリアとして、図8に示されるように、入力文字データ R_0 と L_0 を格納するレジスタ41と42、および暗号化の繰り返し段数 m を指定するレジスタ49が生成される。また、処理の開始と停止を通知するSTART/STOP信号を格納するレジスタ47、処理の終了を表すENDフラグを通知するレジスタ48、およびDESの暗号化が完了した文字データが格納されるレジスタ45、46も生成される。

【0064】一方、内部回路としては、DESの暗号化アルゴリズムを実現するために、 m 個の暗号化鍵 K_1 、 K_2 、 \dots 、 K_m が設定されるレジスタ43-1、43-2、 \dots 、43- m 、DES関数発生器44-1、44-2、 \dots 、44- m 、クロック回路50、減算カウンタ51、およびOR回路52が定義される。

【0065】また、図9に示されるように、暗号化鍵に対応する乱数を発生する乱数発生器53、縮約転置を行うためのビット圧縮回路54、巡回シフトを行うためのビットシフター55-1、55-2、 \dots 、55- m 、56-1、56-2、 \dots 、56- m 、およびシフト後の暗号化鍵を格納するレジスタ57-1、57-2、 \dots 、57- m も定義される。

【0066】実行フェーズでは、繰り返し段数 m をレジスタ49に設定し、入力文字データ R_0 と L_0 をレジスタ41と42に設定し、START/STOP信号をSTARTにすると、クロック回路50によりクロック信

号が発生し、減算カウンタ51、DES関数発生器44-i、乱数発生器53、ビット圧縮回路54、ビットシフター55-i、56-i ($i=1, \dots, m$) にクロック信号が伝わる。

【0067】減算カウンタ51は、繰り返し段数 m だけ数えると値0を出力し、OR回路52の出力するHOLD信号は1から0になる。これにより、クロック回路50と減算カウンタ51は停止する。

【0068】ビット圧縮回路54は、乱数発生器53が発生する乱数のビット長を削減し、ビットシフター55-i、56-iは、クロック信号に同期して、圧縮された乱数をシフトし、レジスタ57-iに入力する。また、DES関数発生器44-iは、クロック信号に同期して、逐次パイプライン構造により、暗号化鍵 K_i を用いた一連の計算を行い、 m クロックサイクル後に暗号化が完了した文字データをレジスタ45、46に出力する。

【0069】DESの復号化回路は、構成としては図8および図9の暗号化回路と同様であり、暗号化が完了した文字データを入力として、平文の文字データを出力する。図10は、RSAの暗号化回路の例を示している。RSAによる暗号化のコンフィギュレーションフェーズでは、まず入力/出力のエリアとして、公開暗号化鍵 e を格納するレジスタ61と、公開暗号化鍵 n を格納するレジスタ62と、入力としての平文 M を格納するレジスタ

$$\begin{aligned} M^0 &= 1 \\ M^1 &= M \pmod{n} \\ M^2 &= M^1 \cdot M \pmod{n} \\ &\dots\dots\dots \\ M^{e-1} &= M^{e-2} \cdot M \pmod{n} \\ M^e &= M^{e-1} \cdot M \pmod{n} \end{aligned}$$

(5) 式の右辺はすべて、法 n による除算の剰余を表している。図10の乗算器69と剰余器70は、HOLD信号が0になるまで(5)式に基づく計算を実行し、最終的に $M^e \pmod{n}$ を出力する。こうして、暗号文 C が生成される。

【0074】また、RSAによる復号化のコンフィギュレーションフェーズでは、図11のような復号化回路が構成される。図11の回路は、図10の暗号化回路と同様の構成であるが、暗号化鍵 e 、 n と平文 M の代わりに、復号化鍵 d 、 n と暗号文 C が、それぞれレジスタ61、62、63に入力されることになる。また、レジスタ71からは、暗号文 C の代わりに、平文 M が出力される。実行フェーズにおける復号化回路の動作は、図10の暗号化回路と同様である。

【0075】次に、図12から図15までを参照しながら、本発明の暗号化/復号化装置の適用例について説明する。図12は、暗号化/復号化装置の仕様を定期的に更新する方法を示している。図12において、プログラマブル論理素子/装置30は、ネットワーク40を介し

タ63が生成される。また、処理の開始と停止を通知するSTART/STOP信号を格納するレジスタ64、処理の終了を表すENDフラグを通知するレジスタ65、およびRSAの暗号化が完了した暗号文 C が格納されるレジスタ71も生成される。

【0070】一方、内部回路としては、RSAのアルゴリズムを実現するために、クロック回路66、減算カウンタ67、OR回路68、乗算器69、および剰余器70が定義される。

10 【0071】実行フェーズでは、暗号化鍵 e をレジスタ61へ、暗号化鍵 n をレジスタ62へセットし、平文 M をレジスタ63に設定し、START/STOP信号をSTARTにすると、クロック回路66によりクロック信号が発生し、減算カウンタ67、乗算器69、剰余器70にクロックが伝わる。

【0072】減算カウンタ67は、暗号化鍵 e の値だけ数えると値0を出力し、OR回路68の出力するHOLD信号は1から0になる。これにより、クロック回路66と減算カウンタ67は停止する。減算カウンタ67

20 は、最大 n まで数えられるようになっている。

【0073】乗算器69と剰余器70は、クロック信号に同期して、(3)式に相当する一連の計算を行う。

(3) 式の右辺の $M^e \pmod{n}$ は、 M^e を n で除算したときの剰余と解釈することができるが、これは次式のような展開式により求めることができる。

(5)

て、遠隔地のコンピュータ81と結ばれている。ノード82、83は、ネットワーク40上に設けられた中継コンピュータや中継器などである。

【0076】コンピュータ81はタイマを用いて時間を計測し、一定時間毎に、設定データの変更指示をホストCPU21に送信する。これにより、ホストCPU21は、設定データを変更して、コンフィギュレーションを再度実行し、暗号化/復号化装置の機能を変更する。このとき、例えばアルゴリズムや鍵が更新される。遠隔地のコンピュータ81の代わりに、ホストCPU21や他のスタンドアロンのCPUで、時間を計測してもよい。

40 【0077】このようなシステムにより、一定時間毎に暗号化/復号化装置の仕様を更新することができ、暗号がより解読されにくくなる。図13は、図12のシステム構成において、暗号化/復号化装置の仕様を外部からの要請に基づいて更新する方法を示している。図13において、ホストCPU21からの接続要求が遠隔地のコンピュータ81に伝えられると、コンピュータ81による接続許可の返信時に、設定データの変更を指示する。

これにより、設定データが変更され、コンフィギュレーションが再度実行されて、暗号化／復号化装置の機能が変化する。

【0078】このようなシステムにより、暗号化／復号化装置の仕様を、外部から更新することができるようになる。図14は、暗号化／復号化装置の仕様を被暗号化データの機密度に応じて更新する方法を示している。図14において、プログラマブル論理素子／装置30は、ネットワーク40を介して、遠隔地のコンピュータ85、87、89、91と結ばれている。ノード84、86、88、90、92は、ネットワーク40上に設けられた中継コンピュータや中継器などである。

【0079】この暗号化／復号化装置では、安全性をより高めるために、あらかじめアルゴリズムや鍵を複数用意しておき、データの転送経路や要求される機密度によって、使用するアルゴリズムや鍵の種類を変化させる。

【0080】ここでは、ホストCPU21は、コンピュータ85との通信にRSAアルゴリズムと暗号化鍵e1を使用し、コンピュータ87との通信にDESアルゴリズムと暗号化鍵k1を使用し、コンピュータ89との通信にRSAアルゴリズムと暗号化鍵e2を使用し、コンピュータ91との通信にDESアルゴリズムと暗号化鍵k2を使用している。

【0081】このようなシステムにより、通信経路の安全性やデータの機密度などに応じて、暗号化／復号化装置の仕様を変更することができ、暗号がより解読されにくくなる。

【0082】図15は、暗号化／復号化装置の仕様を、必要とされる処理速度に応じて変更する方法を示している。図15において、プログラマブル論理素子／装置30、30'は、ネットワーク40を介して、遠隔地のコンピュータ93と結ばれている。ノード94、95、96は、ネットワーク40上に設けられた中継コンピュータや中継器などである。プログラマブル論理素子／装置30にはホストCPU21が接続され、プログラマブル論理素子／装置30'にはホストCPU21'が接続されている。

【0083】被暗号化データが、例えば画像のように大量に存在するデータである場合、データ量と要求される処理速度とに応じて、鍵のビット長やDESアルゴリズムの繰り返し段数などを変化させる。これにより、プログラマブル論理素子／装置30による小規模な画像データの処理と、プログラマブル論理素子／装置30'による大規模な画像データの処理とを、一定時間内で終了させることができる。

【0084】また、図12から図15までのシステムにおいて、暗号化／復号化装置の仕様を変更するために必要な変更データを暗号化して、遠隔地のコンピュータ81、85、87、89、91からホストCPU21、21'に送信することもできる。

【0085】この場合、遠隔地のコンピュータ81、85、87、89、91には、例えば本発明の暗号化／復号化装置が接続され、それにより変更データが暗号化される。また、受信された暗号化変更データはプログラマブル論理素子／装置30、30'により復号化され、その内容に基づいて仕様変更される。このように、変更データを暗号化してやり取りすることで、仕様変更の事実や更新後の仕様を他人に知られることが防止される。

【0086】以上説明した実施形態においては、主としてDESアルゴリズムとRSAアルゴリズムを用いた例を説明したが、これらは一例に過ぎず、本発明の暗号化／復号化装置は、他の任意の暗号化／復号化アルゴリズムを実装することができる。また、設定データの内容はアルゴリズムに応じて変換し、例えば、使用するハードウェア記述言語ライブラリ26のファイル名を、設定データに直接記述することも可能である。

【0087】

【発明の効果】本発明によれば、高速かつフレキシブルな暗号化／復号化装置が実現される。これにより、大規模なデータの暗号化／復号化装置やリアルタイムの暗号化／復号化装置を、機密の度合いや用途に応じてエンドユーザがカスタマイズしたり、自動生成したりすることが可能になる。

【図面の簡単な説明】

【図1】本発明の暗号化／復号化装置の原理図である。

【図2】実施形態における暗号化／復号化装置の構成図である。

【図3】情報処理装置の構成図である。

【図4】ライブラリの例を示す図である。

【図5】暗号化アルゴリズムファイルの例を示す図である。

【図6】コンフィギュレーションフェーズにおける処理のフローチャートである。

【図7】実行フェーズを示す図である。

【図8】DESの暗号化回路を示す図（その1）である。

【図9】DESの暗号化回路を示す図（その2）である。

【図10】RSAの暗号化回路を示す図である。

【図11】RSAの復号化回路を示す図である。

【図12】仕様の定期的な更新方法を示す図である。

【図13】要請に基づく仕様の更新方法を示す図である。

【図14】機密度に応じた仕様の変更方法を示す図である。

【図15】処理速度に応じた仕様の変更方法を示す図である。

【図16】DESのアルゴリズムを示す図である。

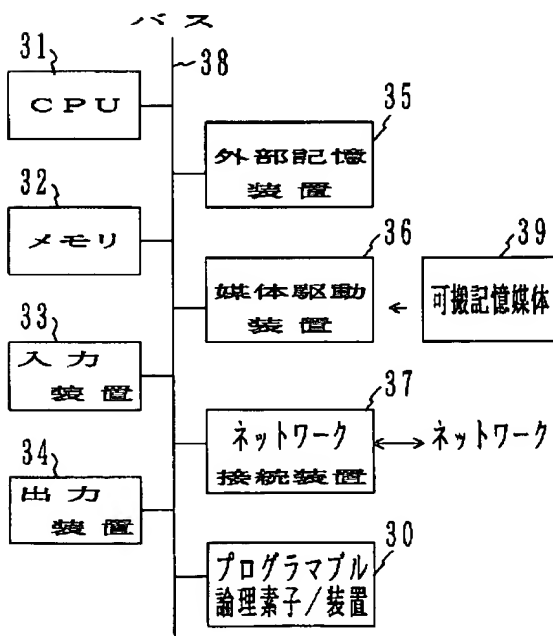
【図17】RSAのアルゴリズムを示す図である。

【符号の説明】

- 1、3 縮約転置
- 2 巡回シフト
- 4、7 転置
- 5 非線形変換
- 6 加算
- 8 暗号化
- 9 復号化
- 11 回路手段
- 12 変更手段
- 13 プログラマブル論理素子
- 20、38 バス
- 21、21' ホストCPU
- 22 外部装置またはネットワーク
- 23 データベース
- 24 暗号化/復号化アルゴリズムファイル
- 25 ハードウェア記述言語ライブラリ生成装置
- 26 ハードウェア記述言語ライブラリ
- 27 ハードウェア記述言語コンパイラ
- 28 マッピングデータオブジェクト
- 29 コンフィギュレーション装置
- 30、30' プログラマブル論理素子/装置
- 31 CPU
- 32 メモリ
- 33 入力装置
- 34 出力装置

【図3】

情報処理装置の構成図 ライブラリの例を示す図



```

module Bcount16(q, clk)
  output [15:0] q;
  input clk;
  reg [15:0] q;
  always@(posedge clk)
    q=q+ d1;
endmodule

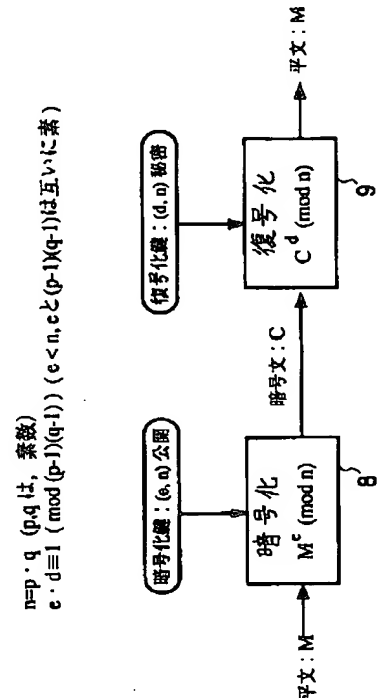
```

【図4】

- 35 外部記憶装置
- 36 媒体駆動装置
- 37 ネットワーク接続装置
- 39 可搬記憶媒体
- 40 ネットワーク
- 41、42、43-1、43-2、43-3、43-m、45、46、47、48、49、57-1、57-2、57-3、57-m、61、62、63、64、65、71 レジスタ
- 10 44-1、44-2、44-3、44-m DES関数発生器
- 50、66 クロック回路
- 51、67 減算カウンタ
- 52、68 OR回路
- 53 乱数発生器
- 54 ビット圧縮回路
- 55-1、55-2、55-3、55-4、55-m、56-1、56-2、56-3、56-4、56-m ビットシフター
- 20 69 乗算器
- 70 剰余器
- 81、85、87、89、91 コンピュータ
- 82、83、84、86、88、90、92、94、95、96 ノード

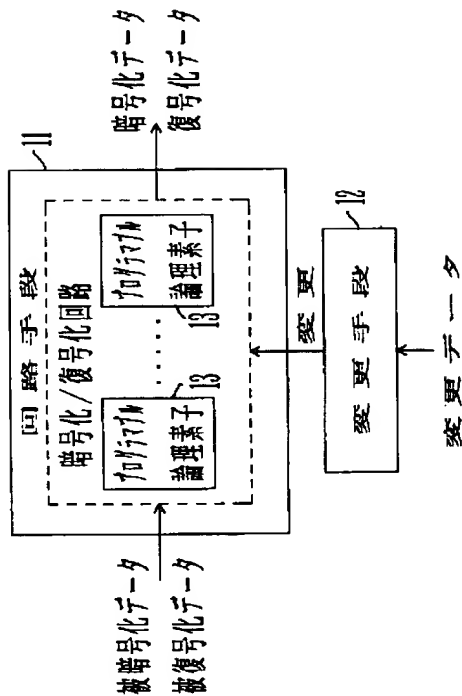
【図17】

RSAのアルゴリズムを示す図



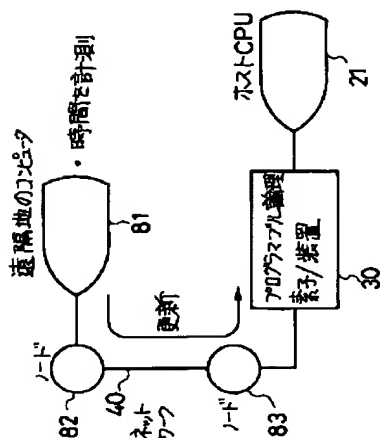
【図1】

本発明の原理図



【図12】

仕様の定期的な更新方法を示す図



【図5】

暗号化アルゴリズムファイルの例を示す図

```

module top;
  reg clock, reset, start, end;
  L1 ~ wire [b1:0] M, C;  <-15
  L2 ~ wire [b2:0] e;    <- 7
  L3 ~ wire [b3:0] n;    <-63

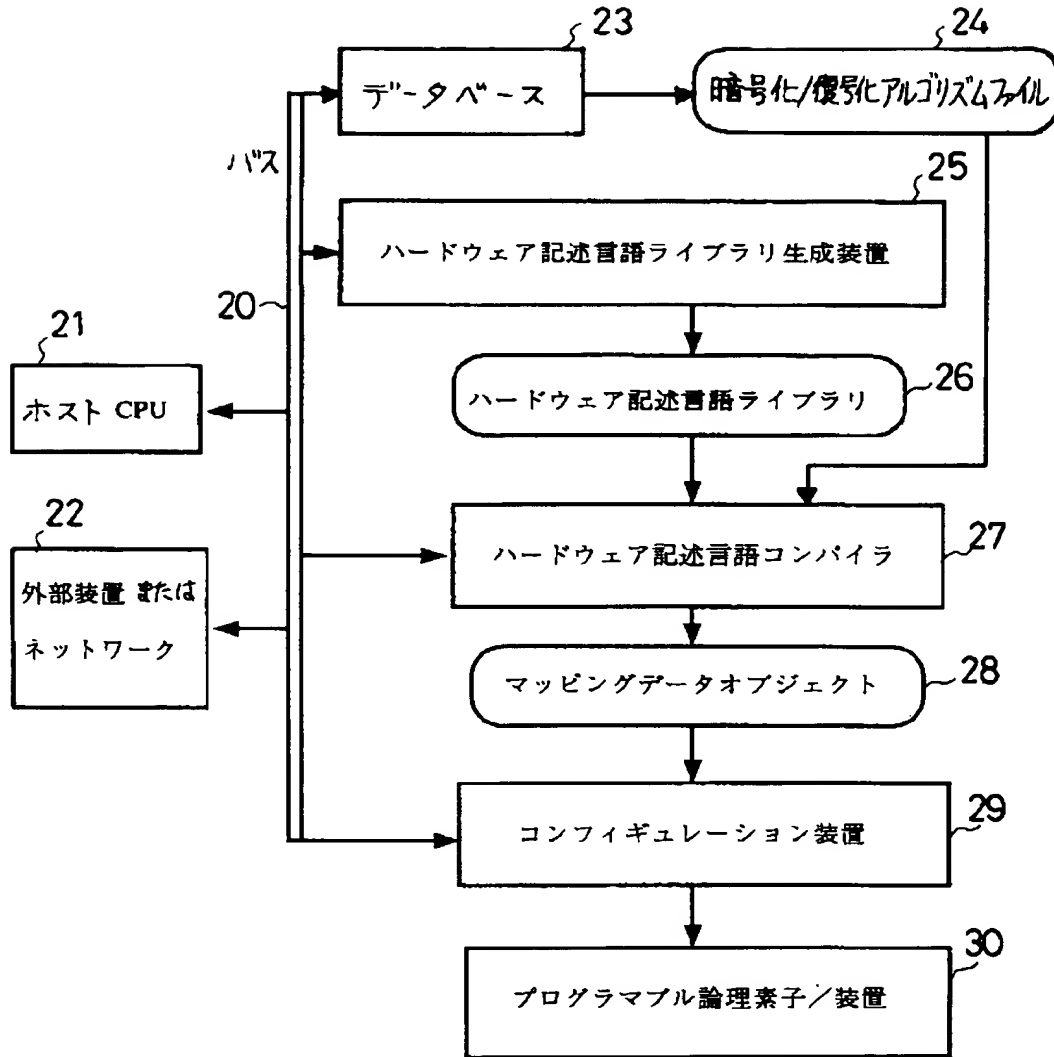
  rsaEnc  enc1(M, C, e, n, clock, reset, start, end);
endmodule

module rsaEnc(M, C, e, n, clk, res, st, ed);
  input [b1:0] M;  <-15
  input [b2:0] e;  <- 7
  input [b3:0] n;  <-63
  input clk, res, st;
  output [b1:0] C; <-15
  output ed;
  {
    integer i;
    always@(posedge clk)
      if (res == 1 `b1)
        C = 16 `d0;
      else if (st == 1 `b1)
        C = 1 `b1;
        for (i=0; i<e; i++) {
          C=(M*C)%n;
        }
        ed=1 `b1;
  }
endmodule

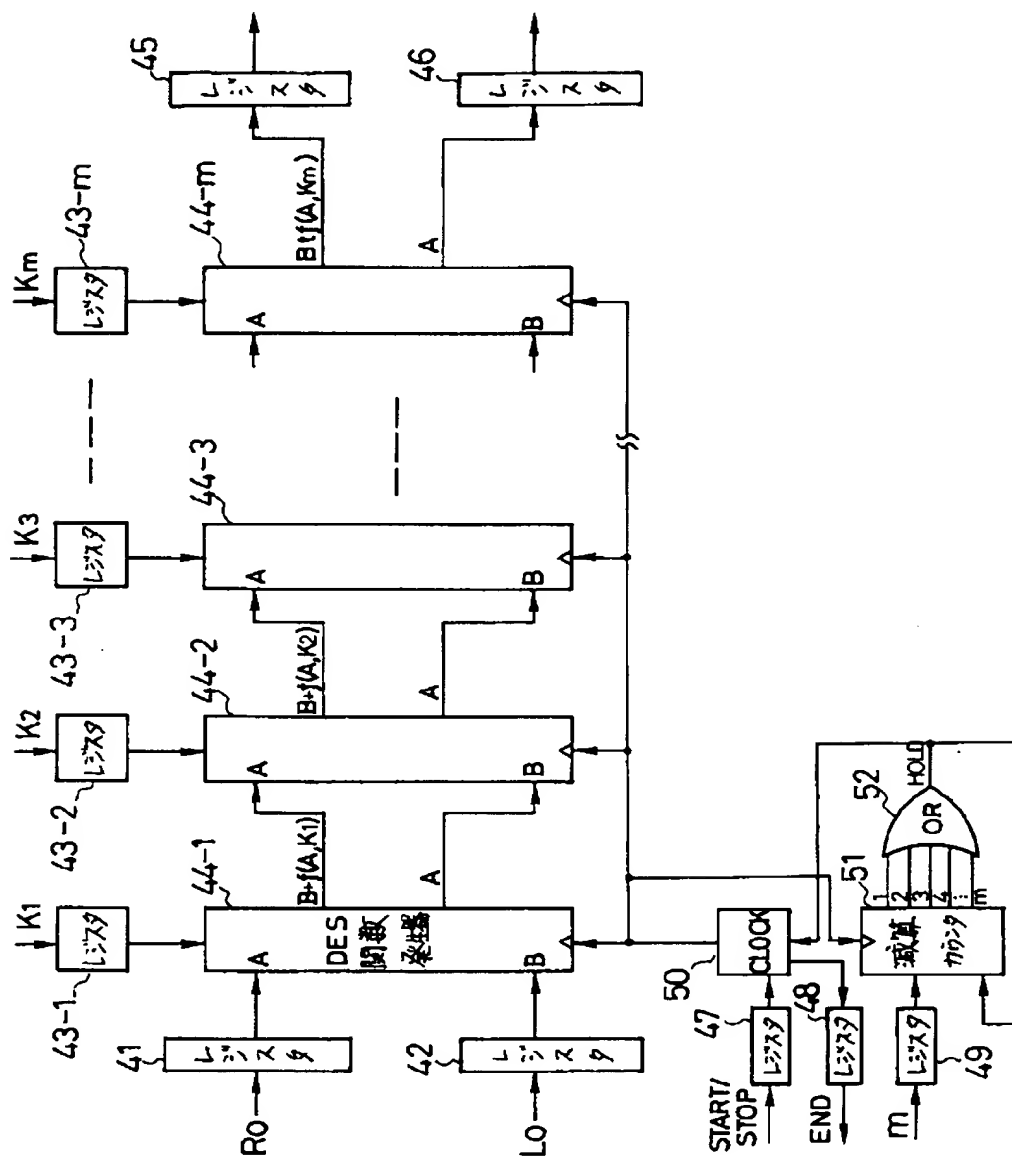
```

【図2】

暗号化/復号化装置の構成図

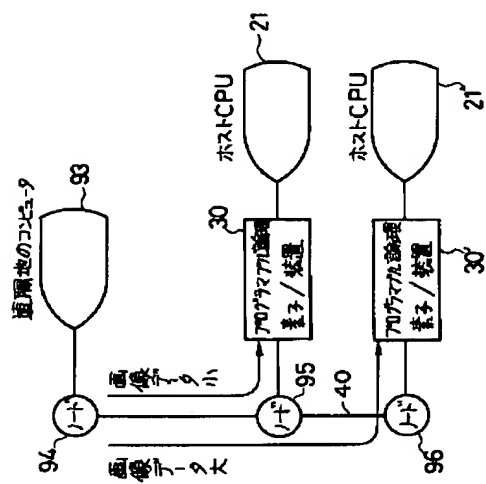


DESの暗号化回路を示す図 (その1)



【図 15】

処理速度に応じた仕様の変更方法を示す図



【図16】

DESのアルゴリズムを示す図

